Chapter 8. Web Services Security

Security is one of the key issues that developers of web services face, particularly in the enterprise. Without a comprehensive security infrastructure, web services will simply not reach their highest potential. It is no surprise that we are starting to see new battles emerge in the marketplace as companies vie for the dominant security position.

Authentication is one of the key components that has emerged. Currently, there are three widely known, competing (and unfortunately, incompatible) web service authentication infrastructures jockeying for position in the marketplace:

Passport

Microsoft's proprietary single sign-on service that provides authentication and digital wallet services for millions of users.

Magic Carpet

AOL's own single sign-on service and digital wallet for use by AOL members.

Sun's Liberty Project

A collaborative effort among Java and open source development communities to develop an alternative to Passport.

Of the three, Passport is the best known and understood architecture. We discuss that architecture in this chapter, but first we will look more closely at web services security in general, including a look at the XML digital signature and XML encryption specifications.

8.1 What Is a "Secure" Web Service?

Web services are all about moving information; it doesn't really matter what type of information is being moved. A "secure" web service is one in which the information sender trusts that the recipient of that information is really who he claims to be and vice versa. Also, a "secure" web service is one in which the information can be received and accessed only by the intended recipient. This definition implies two things:

1. There must be some type of authentication.
2. There must be some type of privacy and integrity protection, such as encryption and authorization.

8.1.1 Authentication

Authentication asks questions like:

- Who am I?
- How do I prove who I am?
- Why should you trust me when I tell you who I am?

- Who are you?
- How can I prove that you are who you say you are?
- Why should I trust you when you tell me who you are?

In the web services world, answering these questions is vitally important. Of equal importance is coming up with a standard *method* to ask and answer these questions.

That's where protocols like the Security Assertions Markup Language (SAML) come into play. In Chapter 5 we briefly discussed SAML and demonstrated in a very simple way how it can be used to provide single sign-on capabilities, but there is more to it than that.

SAML assertions can provide a standard machine-readable expression of a person, application, or device's identity. This identity can be validated, passed around, and used as proof that you really are who you say you are. Because the assertion is digitally signed, we can establish some sort of trust based not on my word that I am who I say I am, but on the word of a trusted third party (the issuer of the assertion).

Even though SAML still has hurdles to jump through on its way to completion, there is a huge amount of potential to provide a very comprehensive, standard framework for implementing global sign-on.

Microsoft recently proposed an alternative approach to authentication based on embedding structures such as Kerberos tickets within the SOAP header. This approach, defined by two related specifications called WS-Security and WS-License, is used extensively by Microsoft's .NET My Services project (formerly known as Hailstorm).

There are no standards (real or de facto) that define how to carry authentication information within a SOAP Envelope.

8.1.2 Privacy

There are two important issues involved in ensuring privacy; both address the protection of assets. The first issue is the protection of an individual's personal information. For example, if I give you my home address and credit card number, I expect that you will protect that information and not send it out over the Internet unguarded. The second issue deals with how you would actually go about protecting that data; this aspect of protection implies authorization policies that detail who is allowed access to the information and what they are allowed to do with it, and encryption methods that ensure unauthorized parties cannot access it.

Currently, there is little Internet privacy infrastructure. While there are simple mechanisms in place to obscure information as it passes from one point to the next (SSL, for example), there is no way to ensure that your personal information is used only for the specific purpose you intended it for once it is sent out over the wire.

The closest thing we have to a privacy infrastructure is the W3C Platform for Privacy Preferences (P3P), which specifies an XML-based language for creating privacy profiles. Service providers create these profiles to tell service consumers how they intend to use the personal information provided by the consumer. While it is a valuable first step, these profiles are not legally binding, can change at any time, and often do. So if a company decides to use

your information in a way that violates the original terms of the profile, there are no laws to stop them. This is changing, though, and hopefully laws will be passed in the very near future to make privacy policies legally binding.

Another problem with privacy on the Internet is that people conflate authentication with authorization. This is seen in the version of Microsoft's Passport service currently being used at a wide range of e-commerce sites. When a user authenticates with Passport, almost all of the personal information contained in his Passport profile is shared automatically with the Passport-enabled web sites the user visits. The problem of authenticating users is solved, but the authorization of companies allowed to access your information is not.

Authentication of a user's identity and management of a user's personal information need to be completely separated from one another. Services like Passport, Magic Carpet, and Liberty blur these lines.

8.2 Microsoft Passport, Version 1.x and 2.x

The current version of Microsoft Passport is designed around providing single sign-on and digital wallet services for web browser-based services. It is easy to understand:

1. A user, Jane, visits a Passport-enabled web site (*MSN.com*, for instance).
2. She is presented with a "Passport Sign-on" link that redirects her web browser to http://www.passport.com/, where she types her Passport user ID and password.
3. Upon validating the username and password, *Passport.com* creates a cookie on Jane's computer that contains her encrypted user profile (all of her personal information).
4. Passport then redirects her back to the original site, which checks for the existence of the cookie, accesses it, and extracts the information it needs about Jane to provide more personalized service.

8.2.1 Drawbacks

There are several problems with this architecture. First, it would be very simple for a malicious person to fool Jane into voluntarily compromising her user ID and password, by simply creating a fake Passport login page. The average user, redirected to the fake page rather than the authentic *Passport.com* page, would not be able to tell the difference. She would enter her login information, press Submit, and never know that she never actually logged in. Meanwhile, the bad guy now has Jane's password and can access all of her personal information and even pretend to be Jane at real Passport-enabled sites.

A second problem is that there is nothing to stop a malicious person from setting up a real Passport-enabled site and taking advantage of Passport, freely dispersing Jane's personal information when she happens to visit the site.

In either situation, it is likely that neither Jane nor Passport will ever know that her information has been compromised because Passport does not include any auditing capabilities that would let Jane go back and monitor the activity of her account.

Another big problem is the natural insecurity of using cookies to store profile information—even if it is encrypted. All it would take is a simple worm virus targeted at locating and

decrypting these encrypted Passport cookies to cause a very serious security issue for the millions of Passport users.

In fact, Microsoft's Passport service has recently come under heavy criticism due to a very serious security flaw that allowed credit card numbers and other personal information stored by Passport to be read by a malicious hacker.

8.3 Microsoft Passport, Version 3.x

While the details are still sketchy, Microsoft is busy working to implement the next generation of their Passport service—this time basing it on the much more secure Kerberos authentication scheme and providing more robust privacy controls.

8.3.1 Overview of Kerberos

Kerberos is an authentication protocol that's been around for quite some time. Originally developed by a university, many large companies, such as Microsoft and IBM, have picked up Kerberos and incorporated support for it into their product lines. Microsoft is by far the largest proponent of Kerberos in the industry today.

Kerberos is too complex to explain in detail. Here's a very abbreviated rundown of how it works:

- First, the user (we'll use Jane again) asks the Kerberos authentication server to validate her credentials. Jane does this by encrypting a packet of information using her private key. The Kerberos server decrypts this packet using her public key. If the decryption was successful, the Kerberos service rules that Jane really is Jane, and sends her an authentication ticket.
- Whenever Jane wants to use some network resource, she must go to the Kerberos Ticket Granting Service (TGS) and explicitly ask permission to use that specific service. The TGS will validate her authentication ticket and issue, as appropriate a one-time use ticket for the service she is requesting.
- Jane presents that one-time use ticket to the service when she submits her request. Also included in the request is an authenticator that proves the one-time use ticket is authentic and really did come from Jane, not some malicious bad guy trying to impersonate her.

That, while being a gross oversimplification, is all that Kerberos does. Passport 3.0 will implement this model, allowing (hopefully) a more robust and secure authentication model that will offer better protection to the services' 160 million plus users.This process protects the user from the impersonation and spoofing attacks that are possible in the current version of Passport. It is a huge advantage.

The Passport privacy management is also improved. Passport users will be able to establish policies that dictate how their information is shared, and who is allowed to access it. The details have not been fleshed out yet, but it will use P3P privacy policies. This doesn't change the fact that P3P policies are not yet legally binding, but it is a huge leap forward for Passport.

8.4 Give Me Liberty or Give Me ...

A new arrival in the web services security battle is a collaborative project called Liberty. Sponsored by Sun and a handful of significant industry players, this project seeks to achieve three main objectives:

1. To allow individual consumers and businesses to maintain personal information securely by enabling a decentralized approach to garnering personal or proprietary information, and promoting interoperability or service delivery across networks.
2. To provide a universal, open standard for "single sign-on," which users and service providers can rely upon, and leverage to interoperate.
3. To provide an open standard for network identity spanning all network-connected devices, allowing the providers of network services and the infrastructure that enables those services to adopt a neutral, open standard, available wherever the Internet is available, securing reliable identity authentication across handsets, automobiles, credit cards—literally any device attached to the Internet.

No technical details have been released on how these goals will be met. At the time of this writing, the Liberty project is essentially vaporware.

8.5 A Magic Carpet

There is as little information about AOL's Magic Carpet proposal as there is about Liberty. The little that is available points to Magic Carpet being an extension of AOL's Screen Name service. Screen Name attempts to provide a single sign-on that can give access to many different web sites across the Web. You can create a profile for the web sites you visit, and you can limit the web sites' access to various parts of your profile. At the time of this writing, AOL's Magic Carpet is still in stealth mode, and should be considered vaporware like Sun's Liberty.

8.6 The Need for Standards

Microsoft is not the only company working on this problem. Unfortunately, those who are working on it are not necessarily working together. To date, we have three incompatible solutions being proposed for web services. In a battle that is starting to resemble the Great Browser Wars of old, traditional enemies are drawing lines and duking it out over who is going to control web services security.

A better approach (the approach that developers need to demand) is for standards to be developed and adopted by all the different players. While it will probably be a long time coming, what we don't need is to develop a great new open interoperability and integration-focused architecture only to have interoperability break down once we actually try to do something really interesting, like global sign-on.

8.7 XML Digital Signatures and Encryption

Two positive examples of standardization efforts currently going on the XML and web services arena are the XML Digital Signature and XML Encryption activities being conducted primarily through the W3C (the IETF is also involved heavily in the XML Digital Signature

effort). Providing comprehensive digital signature and encryption support will be by far a more important issue than the choice of authentication services.

The XML Digital Signature project is working to define a standard syntax for digitally signing data (including XML data) and for encoding that signature as XML. Digital Signatures are critical to protecting the integrity of business transactions on the Internet and will be a key piece of the overall web services infrastructure.

The XML Encryption project is working to define how encrypted data (including XML data) and the metainformation necessary to decrypt that data can be encoded as XML. Encryption of XML data is critical to ensuring the confidentiality of information exchanged between web service participants.

Currently, only IBM's Web Services ToolKit and Microsoft's .NET web services implementations include support for digitally signing and encrypting SOAP messages. While they both support the same XML Encryption and Digital Signature standards, they have different ideas as to how exactly signatures and encrypted data should be placed within a SOAP Envelope. So while each set of tools supports encryption and signing, they are not compatible with one another.

Expect these interoperability differences to be worked out soon, leading to compatible implementations. A key issue will be getting other web service tool vendors to support these security standards within their products.